NAME : IRFAN ULLAH

ID   : 15431

SUBJECT: OBJECT ORINTED PROGRAMING

FINAL   : SUMMER EXAM

TEACHER: AYUB KHAN

Q NO:-1

How many variables are being supported by java justify your answer with the help java coded example for each variable?

ANSWER:-

Definition:-variable provides us with named storage that our programs can manipulate. Each variable in Java has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.

You must declare all variables before they can be used. Following is the basic form of a variable declaration –

data type variable [= value][, variable [= value] ...] ;
data type is one of Java's datatypes and *variable* is the name of the variable. To declare more than one variable of the specified type, you can use a comma-separated list.

Example:-

```
int a, b, c;          // Declares three ints, a, b, and
c.
int a = 10, b = 10;  // Example of initialization
byte B = 22;         // initializes a byte type
variable B.
double pi = 3.14159; // declares and assigns a value of
PI.
char a = 'a';        // the char variable a iis
initialized with value 'a'
```

Kinds of variable:-

There are three kinds of variables in Java –

- Local variables
- Instance variables
- Class/Static variables

Local Variables:-

- Local variables are declared in methods, constructors, or blocks.
- Local variables are created when the method, constructor or block is entered and the variable will be destroyed once it exits the method, constructor, or block.
- Access modifiers cannot be used for local variables.
- Local variables are visible only within the declared method, constructor, or block.
- Local variables are implemented at stack level internally.
- There is no default value for local variables, so local variables should be declared and an initial value should be assigned before the first use.

Example:-

```
public class Test {
  public void pupAge() {
    int age = 0;
    age = age + 7;
    System.out.println("Puppy age is : " + age);
  }

  public static void main(String args[]) {
    Test test = new Test();
    test.pupAge();
  }
}
```
This will produce the following result –

Output:-

Puppy age is: 7

EXAMPLE 2:-

```java
public class Test {
    public void pupAge() {
    int age;
        age = age + 7;
        System.out.println("Puppy age is : " + age);
    }

    public static void main(String args[]) {
        Test test = new Test();
        test.pupAge();
    }
}
```

This will produce the following error while compiling it −

Output:-

Test.java:4:variable number might not have been initialized
age = age + 7;
    ^1 error

Instance Variables:-

- Instance variables are declared in a class, but outside a method, constructor or any block.
- When a space is allocated for an object in the heap, a slot for each instance variable value is created.
- Instance variables are created when an object is created with the use of the keyword 'new' and destroyed when the object is destroyed.
- Instance variables hold values that must be referenced by more than one method, constructor or block, or essential parts of an object's state that must be present throughout the class.
- Instance variables can be declared in class level before or after use.
- Access modifiers can be given for instance variables.
- The instance variables are visible for all methods, constructors and block in the class. Normally, it is recommended to make these variables private

(access level). However, visibility for subclasses can be given for these variables with the use of access modifiers.

- Instance variables have default values. For numbers, the default value is 0, for Booleans it is false, and for object references it is null. Values can be assigned during the declaration or within the constructor.
- Instance variables can be accessed directly by calling the variable name inside the class. However, within static methods (when instance variables are given accessibility), they should be called using the fully qualified name. *ObjectReference.VariableName*.

Example:-

[Live Demo](#)
```java
import java.io.*;
public class Employee {

  // this instance variable is visible for any child class.
  public String name;

  // salary  variable is visible in Employee class only.
  private double salary;

  // The name variable is assigned in the constructor.
  public Employee (String empName) {
    name = empName;
  }

  // The salary variable is assigned a value.
  public void setSalary(double empSal) {
    salary = empSal;
  }

  // This method prints the employee details.
  public void printEmp() {
    System.out.println("name  : " + name );
    System.out.println("salary :" + salary);
  }
```

```
  public static void main(String args[]) {
    Employee empOne = new Employee("Ransika");
    empOne.setSalary(1000);
    empOne.printEmp();
  }
}
```

This will produce the following result −

Output:-

```
name  : Ransika
salary :1000.0
```

Class/Static Variables

- Class variables also known as static variables are declared with the static keyword in a class, but outside a method, constructor or a block.
- There would only be one copy of each class variable per class, regardless of how many objects are created from it.
- Static variables are rarely used other than being declared as constants. Constants are variables that are declared as public/private, final, and static. Constant variables never change from their initial value.
- Static variables are stored in the static memory. It is rare to use static variables other than declared final and used as either public or private constants.
- Static variables are created when the program starts and destroyed when the program stops.
- Visibility is similar to instance variables. However, most static variables are declared public since they must be available for users of the class.
- Default values are same as instance variables. For numbers, the default value is 0; for Booleans, it is false; and for object references, it is null. Values can be assigned during the declaration or within the constructor. Additionally, values can be assigned in special static initializer blocks.
- Static variables can be accessed by calling with the class name *ClassName.VariableName*.

- When declaring class variables as public static final, then variable names (constants) are all in upper case. If the static variables are not public and final, the naming syntax is the same as instance and local variables.

Example:-

import java.io.*;

public class Employee {

  // salary variable is a private static variable
  private static double salary;

  // DEPARTMENT is a constant
  public static final String DEPARTMENT = "Development ";

  public static void main(String args[]) {
    salary = 1000;
    System.out.println(DEPARTMENT + "average salary:" + salary);
  }
}

This will produce the following result −

Output:-

Development average salary:1000

Note − If the variables are accessed from an outside class, the constant should be accessed as Employee.DEPARTMENT

Q NO 2

Why "If" is used in java justify your answer with the help java coded
example and explain in detail?

Answer:-

Class if statement {

   Public static void main(string[] args){

     int number = 10;

    // checks if number is greater than 0

    If (number>0) {

      System.out.println("the number is positive .");

       }

      System.out.println("this statement is always executed.");

}

    }

Output:-

The number is positive

This statement is always executed

Q no 3

Why "if else if" is used in java justify your answer with the help java coded example and explain in detail?

ANSWER:-

```
Class ladder {

    Public static void main(string[] args) {

        Int number =0;

        // check s if number is greater than 0

        If (number >0) {

            System.out.println("the number is positive.");

    }

        // checks if number is less than 0

        Else if (number<0) {

System.out.println("the number is negative.");

 }

 Else {

System.out.print("the number is 0.");

}

    }

        }
```

Output:-   Number is 0

Q NO 4

What are loops, why they are used in java and how many types of loops are being supported by java explain in detail?

Answer:-

In computer programming, loops are used to repeat a block of code. For example, if you want to show a message 100 times, then rather than typing the same code 100 times, you can use a loop.

In Java, there are three types of loops.

for loop
while loop
do...while loop

```
class Main {

 public static void main(String[] args) {

   int n = 5;

   // for loop

   for (int i = 1; i <= n; ++i) {

     System.out.println(i);

   }

 }

}
```

Q no 5

Write 3's table in decremented form in java which takes input from user write java coded program and explain in detail?

Answer:-

```java
import java.util.Scanner;

public class Main {

 public static void main(String[] args) {
  Scanner in = new Scanner(System.in);
  System.out.println("Input the Number: ");
  int n = in .nextInt();
  for (int i = 1; i <= 10; i--) {
   System.out.println(n + "*" + i + " = " + (n * i));
  }
 }
}
```

Input the number

3*10=30
3*9=27
3*8=24
3*7=21
3*6=18
3*5=15
3*4=12
3*3=9
3*2=6
3*1=3